JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06282443
(43)Date of publication of application: 07.10.1994

(51)Int.Cl.
G06F 9/45
G06F 9/06
G06F 9/06

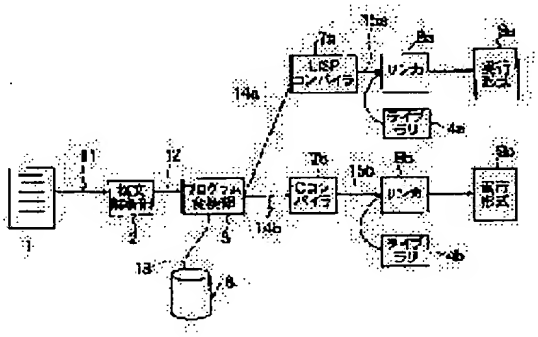
(21)Application number: 05068248
(22)Date of filing: 26.03.1993

(71)Applicant: SONY CORP
(72)Inventor: KITATANI YOSHIMICHI

(54) METHOD AND DEVICE FOR EDITING PROGRAM

(57)Abstract:

PURPOSE: To provide a compilation device and its method which enables a different language to be executed on a different processor element PE processing system with optimum processing efficiency. CONSTITUTION: A syntax analytic part 2 analyzes the syntax of a source program 1 described in the LISP language. A program conversion part 3 generates a program 14a suitable for LISP compilation and a program 14b suitable for C compilation from the syntax analyzed program 11 by referring to the contents of a data base 6. A LISP compiler 7a and a C compiler 7b compile the inputted programs according to their language systems. A linker 8a converts the LISP- compiled program into machine word codes 9a which can be executed on a 1st program processing system by referring to a library 4a. A linker 8b, on the other hand, the C language compiled program into machine word codes 9b which can be executed on a 2nd program processing system by referring a library 4b.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

MENU SEARCH INDEX

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開平6－282443
(43)公開日　平成６年(1994)10月７日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/45				
9/06	４１０　Ｈ	9367－5B		
	４３０　Ｅ	9367－5B		
		9292－5B	Ｇ０６Ｆ　9/ 44　３２２　Ａ	

審査請求　未請求　請求項の数９　ＯＬ　（全 10 頁）

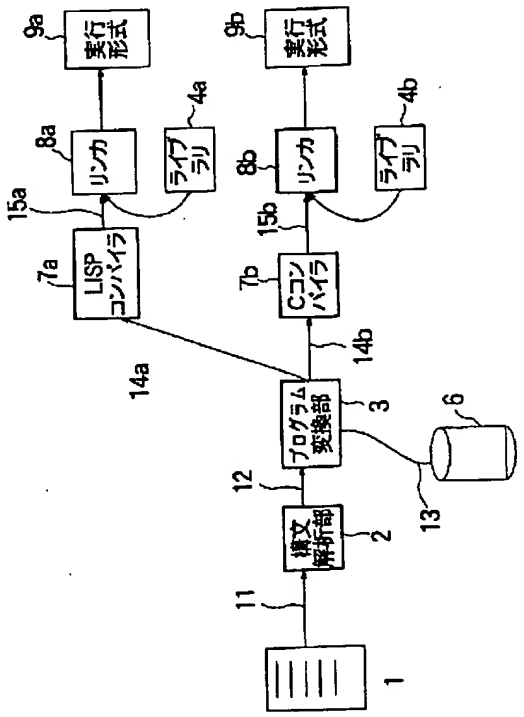
(21)出願番号　特願平5－68248

(22)出願日　平成５年(1993)３月26日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　北谷　義道
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　弁理士　佐藤　隆久

(54)【発明の名称】　プログラム編集方法と装置

(57)【要約】
【目的】　異なる言語をその言語の処理効率を異なるプロセッサエレメントＰＥ処理系で最適に実行することを可能ならしめるコンパイル装置とその方法を提供する。
【構成】　構文解析部２はＬＩＳＰ言語で記述されたソースプログラム１の構文を解析する。プログラム変換部３は構文解析されたプログラム１１をデータベース６の内容を参照して、ＬＩＳＰコンパイルに適したプログラム１４ａとＣコンパイに適したプログラム１４ｂを生成する。ＬＩＳＰコンパイラ７ａおよびＣコンパイラ７ｂはそれぞれ入力されたプログラムをその言語体系にしたがってコンパイルする。リンカ８ａはＬＩＳＰコンパイルされたプログラムをライブラリ４ａを参照して第１のプログラム処理系で実行可能な機械語コード９ａに変換する。リンカ８ｂはＣ言語コンパイルされたプログラムをライブラリ４ｂを参照して第２のプログラム処理系で実行可能な機械語コード９ｂに変換する。



【特許請求の範囲】
【請求項１】高級プログラミング言語で記述されたソースプログラムを、構成部分ごとに処理内容を分析し、それらを複数のプログラム処理系に振り分けそれぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うプログラム編集方法において、
複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行なった結果を記録しているデータベースを用い、
ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索し、
それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換することを特徴とするプログラム編集方法。
【請求項２】ソースプログラム中に定義された一つの関数ブロックを、その中に含まれるプログラム処理に応じて複数のプログラム処理系のどれかに分類し、
該分類されたプログラム処理系に対応するデータベースに登録し、
それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼出しのプログラムに変換することを特徴とする請求項１記載のプログラム編集方法。
【請求項３】プログラム処理系が持つ機能に対してその実行に必要とされるプログラム処理系の資源の多寡を表現する数的データをデータベースに保持し、
そのデータを元に変換の対象となるソースプログラム中の関数呼びだしのシーケンスの実行に要するプログラム処理系の資源を計算し、
計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類し前記データベースに記録することを特徴とする請求項１または２記載のプログラム編集方法。
【請求項４】前記高級プログラミング言語で記述されたソースプログラムは、ＬＩＳＰ言語で記述されている請求項３記載のプログラム編集方法。
【請求項５】高級プログラミング言語で記述されたソースプログラムを、構成部分ごとに処理内容を分析し、それらを複数のプログラム処理系に振り分けそれぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うプログラム編集装置において、
複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行なった結果を記録しているデータベースを有し、
ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索する手段を有し、
それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換する手段を有する、ことを特徴とするプログラム編集装置。
【請求項６】ソースプログラム中に定義された一つの関数ブロックを、その中に含まれるプログラム処理に応じて複数のプログラム処理系のどれかに分類する手段を有し、
分類されたプログラム処理系に対応するデータベースに登録する手段を有し、
それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼出しのプログラムに変換する手段を有することを特徴とする請求項５記載のプログラム編集装置。
【請求項７】プログラム処理系が持つ機能に対してその実行に必要とされるプログラム処理系の資源の多寡を表現する数的データをデータベースに保持し、
そのデータを元に変換の対象となるソースプログラム中の関数呼びだしのシーケンスの実行に要するプログラム処理系の資源を計算する手段を有し、
計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類し前記データベースに記録することを特徴とする請求項５または６記載のプログラム編集装置。
【請求項８】前記高級プログラミング言語で記述されたソースプログラムは、ＬＩＳＰ言語で記述されている請求項７記載のプログラム編集装置。
【請求項９】高級プログラミング言語で記述されたソースプログラムを構成部分ごとに処理内容を分析する構文解析部、
複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベース、
ソースプログラム中で名前によって呼び出す機能を前記データベースから検索し、それらを複数のプログラム処理系に振り分けるプログラム変換部、
それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換する編集手段を有し、それぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うことを特徴とするプログラム編集装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明はデジタル計算機におけるコンパイラに関するものであり、特に複数のプログラム言語が同一の計算機システムで動作する場合におけるコンパイラおよびプログラム編集方法とその装置に関する。
【０００２】
【従来の技術】従来からＣ言語等の高級プログラム言語で記述されたソースプログラムを機械語コードなどの下位言語に変換するコンパイラといったプログラム編集方

法が知られており、実際に使用されている。以下、コンパイラを使用し、高級プログラム言語で記述されたソースプログラムを下位言語に変換する、従来のプログラム編集方法について述べる。

【０００３】図６は従来のコンパイラ装置でのデータの流れを示した図である。このコンパイラ装置は、構文解析部２、コード生成部２３、アセンブラ処理部（アセンブラ）２３、リンカ部２５を有する。図６においてソースプログラム１はＣ言語で記述されたプログラムである。構文解析部２はＣ言語で記述されたソースプログラムの文法チェックなどを行ない、コンパイラの内部表現に変換する。次に、コード生成部２３はその内部データを構文解析部２から受け取り、アセンブラコードを生成する。最後に、アセンブラ２４はコード生成部２３が生成したアセンブラコードを受け取ってコンピュータなどのプログラム処理系において実行可能なファイルを生成する。

【０００４】次に、上記コンパイラ装置によって同じソースプログラム１内や別のソースプログラムをコンパイルすることによって得られた機能ブロックを呼び出すプログラムを処理する場合を考える。このような機能ブロックはソースプログラム中で関数と呼ばれる名前として参照され、呼び出される。構文解析部２は関数呼出しのコードを受け取ると、関数呼出し表す特別なデータ型を生成し、コード生成部２３に渡す。コード生成部２３は関数名に一意に対応するアセンブラコードのラベルを作るルーチンを持つ。このルーチンが関数名に対応するラベルを生成し、そのラベルへのジャンプを表すアセンブラ命令を生成する。

【０００５】さらに、ソースプログラム１内の部分で、関数をブロックを定義している部分を処理する場合を考える。コード生成部２３はその関数機能を表すアセンブラコードを生成する。また先に述べた、関数名に対してアセンブラコードのラベルを生成するルーチンを使ってこの機能ブロックの関数名に対応するラベルを作り、生成したアセンブラコードに付加する。

【０００６】このように作成された機能ブロックの呼出しに対応するアセンブラコードに含まれるラベルの参照は、アセンブラ２４によって機械語コードに翻訳されたあとリンカ２５によってライブラリ２６を参照して実際の番地に変換され、適切な機能ブロックの呼出しコードが生成される。このようにして生成された実行可能な形式のプログラムがプログラム処理系、具体的にはコンピュータにおいて実行される。

【０００７】

【発明が解決しようとする課題】関数の呼出しコードを生成するために、関数名とラベル名だけの情報を持っている上記のコンパイラ装置による上記コンパイル方法では、ソースプログラム中に現れるそれぞれの関数や機能に対応して別々の処理系の機能を呼び出すことは想定されていない。高級プログラミング言語はそれぞれ用途に向き、不向きがあり、例えば、ＦＯＲＴＲＡＮは数値計算に適しているが、計算機のハードウェアに密着した入出力を取り扱うといった用途には向いていない。このため、例えば、ＦＯＲＴＲＡＮを使用して数値演算とＩ／Ｏ操作の混在したプログラムを作成すると、数値計算処理は高速処理可能であるが、Ｉ／Ｏ操作処理は遅いオブジェクトプログラムが生成されるという問題が生じる。

【０００８】したがって、本発明は、ソースプログラムの記述言語に応じた効率のよいオブジェクトプログラムを編集する装置とその方法を提供する。

【０００９】

【課題を解決するための手段、および、作用】上記の問題を解決するために、本発明に係るコンパイラ方法は、それが想定する複数のプログラム処理系、具体的には、コンピュータ、それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベースを用い、ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索し、、それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼出しのプログラムに変換するすることを特徴とする。

【００１０】また、本発明は、ソースプログラム中に定義された一つの関数ブロックを、その中に含まれる処理に応じて複数のプログラム処理系のどれかに分類し、分類されたプログラム処理系に対応するデータベースに登録し、それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼び出しのプログラムに変換することを特徴とする。

【００１１】さらに、本発明は、プログラム処理系が持つ機能に対してその実行に必要とされるプログラム処理系の資源の多寡を表現する数的データをデータベースに保持し、そのデータを元に変換の対象となるソースプログラム中の関数呼び出しのシーケンスの実行に要するプログラム処理系の資源を計算し、計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類しデータベースに記録することを特徴とする。

【００１２】また本発明によれば、上述したプログラム編集方法を実施するプログラム編集装置が提供される。

【００１３】さらに本発明によれば、高級プログラミング言語で記述されたソースプログラムを構成部分ごとに処理内容を分析する構文解析部、複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベース、ソースプログラム中で名前によって呼び出す機能を前記データベースから検索し、それらを複数のプログラム処理系に振り分けるプログラム変換部、それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換する編

集手段を有し、それぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うことを特徴とするプログラム編集装置が提供される。

【００１４】

【実施例】図面を参照して本発明の実施例について説明する。図１は、本発明のプログラム編集方法が適用されるプロセッサエレメントＰＥ（コンピュータシステム）の概略構成図である。図１（Ａ）は単独のプロセッサエレメントＰＥの構成を示す図であり、図１（Ｂ）は複数のプロセッサエレメントＰＥが相互に接続されて大規模コンピュータシステムを構成する図を示す。

【００１５】図１（Ａ）に示したプロセッサエレメントＰＥは、ＬＩＳＰなどの記号言語を処理する記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭと主記憶MMおよび二次記憶装置とを有するストーリジ（記憶装置）から構成されている。記号処理用計算機エバリュエータＥＵに対して、入出力や記憶管理に使われる計算機ＲＭおよびストーリジが一種のリソース（資源）として扱われている。

【００１６】図１（Ｂ）は、各プロセッサエレメントＰＥ内の入出力や記憶管理に使われる計算機ＲＭが、ＬＡＮ（Local Area Network）などの相互接続系（インターリソースコネクション）を介して、相互に接続されて、複数のプロセッサエレメントＰＥが全体として、大規模コンピュータシステムを構成している。

【００１７】図２は図１（Ａ）に示されたプロセッサエレメント（ＰＥ）の構成の詳細を示す図である。本発明のプログラム編集方法は、図２に示すような記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機（ＲＭ）からなる並列計算機ネットワークのプロセッサエレメントＰＥに適用される。

【００１８】記号処理用計算機エバリュエータＥＵは、データ幅３２ビットのＶＬＩＷ（Very Long Instruction Word）制御のプロセッサであり、ユーザ定義の関数やメソッドのコンパイルされたコードを高速実行するためのインストラクションキャッシュ（６４ビット×６４Ｋワード）を持つ。また、記号処理用計算機エバリュエータＥＵは、データのフィールド演算を高速に実現するためのマスカ（Ｍａｓｋｅｒ）／算術論理処理ユニット（ＡＬＵ）を持ち、関数やメソッドの呼び出しを高速化するためのデータフィールド値による多方向分岐機能を持ち、さらに多方向分岐のためのルックアップテーブルと記号処理用計算機エバリュエータＥＵのシステムスタックのための６４Ｋワードの制御用メモリを持つ。さらに記号処理用計算機エバリュエータＥＵ、記号処理計算を実行する際のフレームスタックとレジスタファイルのための６４Ｋワードのローカルメモリを持つ。記号処理用計算機エバリュエータＥＵは、入出力や記憶管理に要する演算は、入出力や記憶管理に使われる計算機ＲＭにその実行を要求する。

【００１９】入出力や記憶管理に使われる計算機ＲＭは、記憶管理、デバイス管理、プロセス管理、ファイル管理、入出力、他のプロセッサエレメントＰＥとの通信の機能を記号処理用計算機エバリュエータＥＵに提供する。入出力や記憶管理に使われる計算機ＲＭには、ワークステーションＷＳを使用して、ワークステーションＷＳのＸウインドウ・システム上にアプリケーション開発時および実行時のインタラクティブなヒューマンインタフェース環境を実現する。そのため、入出力や記憶管理に使われる計算機ＲＭは関数定義やメソッド定義のインクリメンタルな修正・コンパイル、エラー状態からの復帰・実行再開機能を持つ。さらに入出力や記憶管理に使われる計算機ＲＭは、アプリケーションの実行時にユーザとの動的な対話ができるような入出力機能を提供する。入出力や記憶管理に使われる計算機ＲＭは、主記憶MM上の記号処理データの高速アクセスのためのポイント・マニュプレータ（Pointer Manipulator）を持つ。入出力や記憶管理に使われる計算機ＲＭは、並列処理時に他のプロセッサエレメントＰＥとの通信を行なうための通信インタフェースを装着可能である。

【００２０】記憶装置（ストーリジ）のうち主記憶MMは、記号処理用計算機エバリュエータＥＵからも直接アクセス可能なデュアルポートメモリである。主記憶MMでは、排他的アクセス、記号処理演算で有効な参照カウントなどの機能アクセスを提供する。主記憶MMは、これらの制御情報と２ワードのポインタを表現するため、８０ビット×２５６Ｋワードの構成となっている。

【００２１】記憶装置のうち二次記憶は、ワークステーションＷＳのディスクシステムを使用する。

【００２２】記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭとは、入出力や記憶管理に使われる計算機インターフェースＲＭ－Ｉ／Ｆで接続され、記号処理用計算機エバリュエータＥＵから入出力や記憶管理に使われる計算機ＲＭに、また入出力や記憶管理に使われる計算機ＲＭから記号処理用計算機エバリュエータＥＵに機能データを送出する。また、記号処理用計算機エバリュエータＥＵからメインメモリインターフェースMM－Ｉ／Ｆを介して情報データが送出され、主記憶MMと入出力や記憶管理に使われる計算機ＲＭに送出される。

【００２３】入出力や記憶管理に使われる計算機インタフェースＲＭ－Ｉ／Ｆ、および、主記憶インタフェースMM－Ｉ／Ｆは、記号処理用計算機エバリュエータＥＵ内の内部バスＩＢＵＳを介して算術論理処理ユニットＡＬＵに接続される。また、入出力や記憶管理に使われる計算機インタフェースＲＭ－Ｉ／Ｆは、入出力や記憶管理に使われる計算機ＲＭ内のエバリュエータインターフェースＥＵ－Ｉ／Ｆを介してワークステーションＷＳに接続される。

【００２４】図３は本発明のコンパイラ装置のプログラム変換形態を表す図である。このコンパイラ装置は、構文解析部２、プログラム変換部３、ＬＩＳＰコンパイル処理部（ＬＩＳＰコンパイラ）７ａ、Ｃコンパイル処理部（Ｃコンパイラ）７ｂ、ＬＩＳＰ言語に関するリンケージ処理を行う第１のリンカ８ａ、および、Ｃ言語に関するリンケージ処理を行う第２のリンカ８ｂを有する。

【００２５】ソースプログラム１はＬＩＳＰで記述されたソースプログラムである。構文解析部２はＬＩＳＰで記述されたソースプログラムの文法チェックなどを行ない、構文木データなどのコンパイラの内部表現に変換するモジュールである。プログラム変換部３はその内部データを構文解析部２から受け取り、図２に示した記号処理用計算機エバリュエータＥＵおよび入出力や記憶管理に使われる計算機ＲＭで実行されるべきプログラムコードを生成するモジュールである。ＬＩＳＰコンパイラ７ａとＣコンパイラ７ｂはそれぞれ、ライブラリ６ａ、６ｂを参照してプログラム変換部３が生成したプログラムコードを、リンカ８ａ、８ｂを介して、記号処理用計算機エバリュエータＥＵおよび入出力や記憶管理に使われる計算機ＲＭなどのプログラム処理系、つまり、コンピュータで実行可能な機械語コード（オブジェクトプログラム）９ａ、９ｂに変換するコンパイラである。機械語プログラム９ａ、９ｂはコンパイラ７ａ、７ｂが生成した機械語プログラムであり、それぞれ図２の記号処理用計算機エバリュエータＥＵおよび入出力や記憶管理に使われる計算機ＲＭに読み込まれて実行される。データベース６は、図２に示した記号処理用計算機エバリュエータＥＵおよび入出力や記憶管理に使われる計算機ＲＭで実行される関数の情報を蓄積するデータベースであり、プログラム変換部３が関数名をキーとしてその関数が記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭのどちらの処理系で実行されるかを検索するために使われる。

【００２６】以下、本発明のコンパイラ装置の動作について説明する。図４および図５は本発明のコンパイラ装置を用いてプログラム編集する編集方法の処理内容を示すフローチャートである。図４および図５はそれぞれ図３に示したプログラム変換部３におけるコード生成の処理を表し、図４の処理によってＬＩＳＰ言語のコードが生成され、図５のフローチャートに示した処理によってＣ言語のコードが生成される。

【００２７】図４は、図３の構文解析部２からプログラムの解析結果１２を受け取りＬＩＳＰコードを生成する処理を表す。また、プログラム変換部３は解析結果１２が無くなるまで図４の処理を繰り返し実行する。プログラム変換部３は、ステップ０１（Ｓ０１）においてプログラムが関数定義を表すかどうかを判断する。関数定義でなければ、プログラム変換部３は、ステップ０２（Ｓ０２）において式が関数呼出しかどうかを判断する。そうでなければ、プログラム変換部３は、Ｓ０３において通常のＬＩＳＰ処理としてそのままのＬＩＳＰコードを生成し出力する。この出力は図３に示したＬＩＳＰコンパイラ７ａに渡される。Ｓ０１において対象プログラムが関数定義であると判断すると、構文解析部２は、Ｓ０５でそのプログラムを解析する。そしてＳ０６において、プログラム変換部３は、そのプログラムが図２の記号処理用計算機エバリュエータＥＵで実行されるＬＩＳＰコード１４ａとして出力するか、あるいは図２の入出力や記憶管理に使われる計算機ＲＭで実行されるＣコード１４ｂとして出力するかを決定する。ＬＩＳＰコード１４ａとして出力する場合、プログラム変換部３は、Ｓ０７の処理でそのプログラムはＬＩＳＰコード１４ａとしてＬＩＳＰコンパイラ７ａに対して出力する。Ｃコード１４ｂとして出力する場合、プログラム変換部３は、Ｓ０９で図５のＣコード生成処理を呼出しＣコード１４ｂに変換する。

【００２８】Ｓ０２においてプログラム変換部３が関数呼出し式を認識すると、プログラム変換部３は、Ｓ１０において呼びだされる関数が記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭのどちらの処理系に属しているかを判断する。このとき、プログラム変換部３は、図３のデータベース６を検索する。もし入出力や記憶管理に使われる計算機ＲＭに属している関数であれば、プログラム変換部３は、Ｓ１１においてＲＭ関数を呼び出すためのＬＩＳＰコード１４ａを生成する。もし記号処理用計算機エバリュエータＥＵに属している関数であれば、プログラム変換部３は、Ｓ１２においてＬＩＳＰ関数呼出しの形のままＬＩＳＰコード１４ａとして出力する。

【００２９】図３のプログラム１２として渡される構文解析プログラムは多くの部分プログラムから構成されていることがあるので、Ｓ０３とＳ０７でそれに対するコードを生成する場合に、部分プログラムすべてに対して同様に図４の処理を再帰的に行なう。

【００３０】図５の処理内容は、図４を参照して述べたＬＩＳＰコード１４ａを生成する処理と同様であり、ＬＩＳＰコード１４ａを生成する処理を示した図４の処理と同様の処理をＣコード１４ｂを生成するために行ない、Ｃ言語コード１４ｂを出力する。

【００３１】以上の動作によって得られた次段階プログラムはそれぞれＬＩＳＰ用コンパイラ７ａ、Ｃ用コンパイラ７ｂ、および、リンカ８ａおよびリンカ８ｂによって機械語プログラム９ａ～ｂに変換される。以上によって、図３のソースプログラム１に対してその機能によって分類された記号処理用計算機エバリュエータＥＵ、入出力や記憶管理に使われる計算機ＲＭそれぞれで実行される機械語プログラム９ａ、９ｂが得られる。これらは、このプログラムにより記号処理用計算機エバリュエータＥＵおよび入出力や記憶管理に使われる計算機ＲＭ

は交互にお互いの関数を呼び合い処理を実行する。

【００３２】上述した実施例は、図１（Ａ）に示したコンピュータシステム、つまり、記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭとによって構成される単体のプロセッサエレメントＰＥにおける処理を示したが、本発明の実施に際しては、図１（Ｂ）に図解した複数のプロセッサエレメントＰＥで構成される大規模なコンピュータシステムにおいて、あるプロセッサエレメントＰＥの記号処理用計算機エバリュエータＥＵと他のプロセッサエレメントＰＥの入出力や記憶管理に使われる計算機ＲＭとで上述のように生成された機械語コードを実行させることもできる。

【００３３】

【発明の効果】以上述べたように本発明のプログラム編集装置およびその方法によれば、一種類の高級プログラミング言語で記述されたソースプログラムについて、ソースプログラムの部分毎の処理内容を分析し、その処理が呼び出す機能に応じて適切な処理系を選択し、またその処理が定義する機能を適切な処理系に分類し、処理内容に応じて最適な処理系を選択することで、効率良く高速に処理を行わせることを可能にするプログラム編集装置とその方法が提供される。

【図面の簡単な説明】

【図１】本発明が適用される計算機ネットワークの接続形態を示す図であり、（Ａ）は、プロセッサエレメントＰＥ単体の構成図であり、（Ｂ）は複数のプロセッサエレメントＰＥが大規模コンピュータシステムを構成する図である。

【図２】本発明が適用されるプログラム処理系（計算機ネットワーク）の構成図である。

【図３】本発明のプログラム編集装置におけるプログラム変換方法を示す図である。

【図４】本発明のプログラム編集方法の処理の第１例としてＬＩＳＰコードを生成する処理を示すフローチャートである。

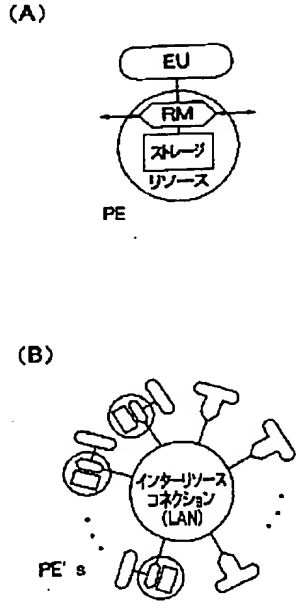
【図５】本発明のプログラム編集方法の処理の第２例としてＣコードを生成する処理を示すフローチャートである。

【図６】従来のプログラム編集装置の構成を表す図である。

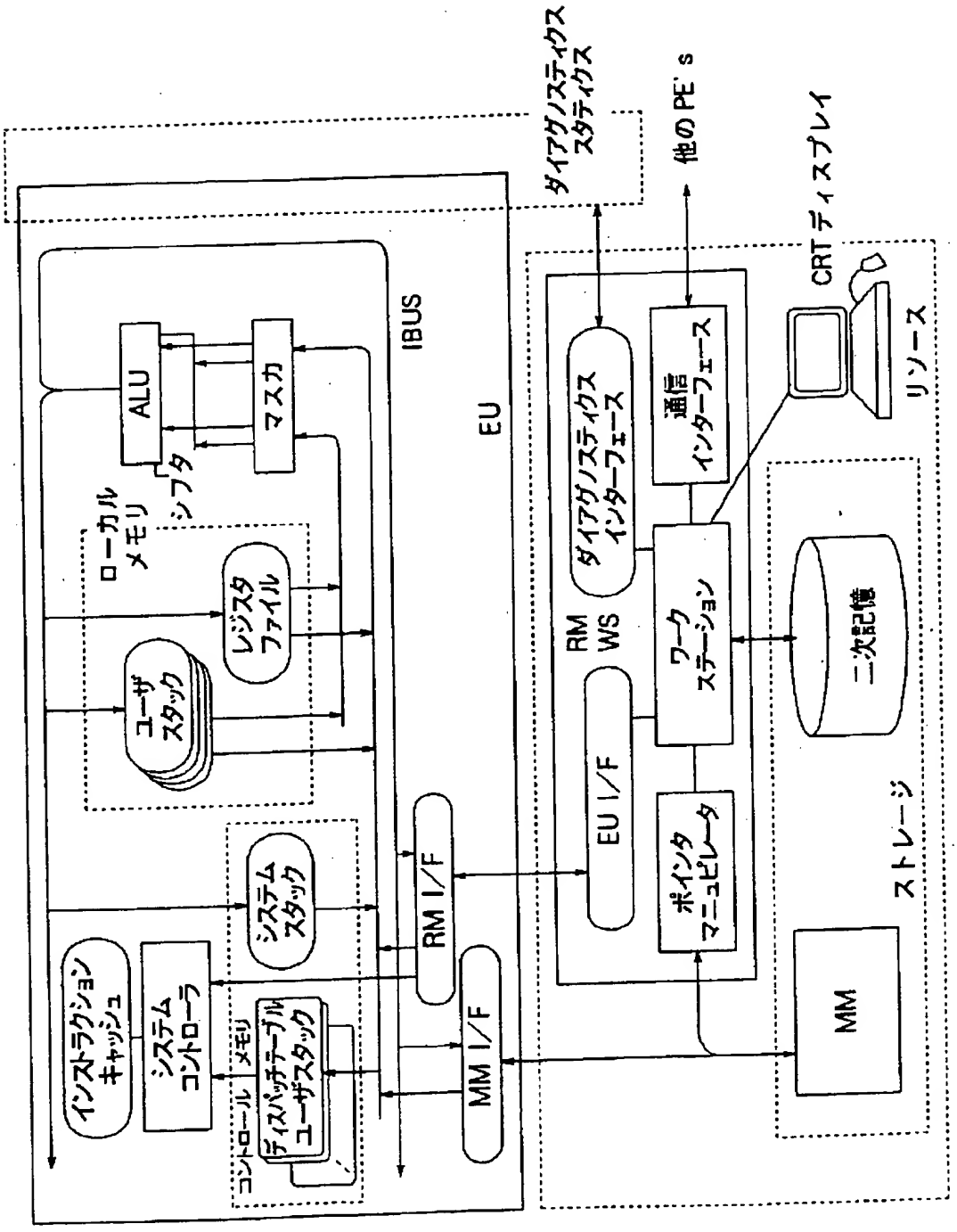
【符号の説明】

１・・ソースプログラム
２・・構文解析部
３・・プログラム変換部
４ａ，４ｂ・・ライブラリ
６・・データベース
７ａ・・ＬＩＳＰコンパイラ
７ｂ・・Ｃコンパイラ
８ａ・・ＬＩＳＰリンカ
８ｂ・・Ｃリンカ
９ａ・・ＬＩＳＰコードによる実行可能な形式のプログラム（機械語コード）
９ｂ・・Ｃコードによる実行可能な形式のプログラム（機械語コード）
ＲＭ・・入出力や記憶管理に使われる計算機
ＥＵ・・記号処理用計算機エバリュエータ

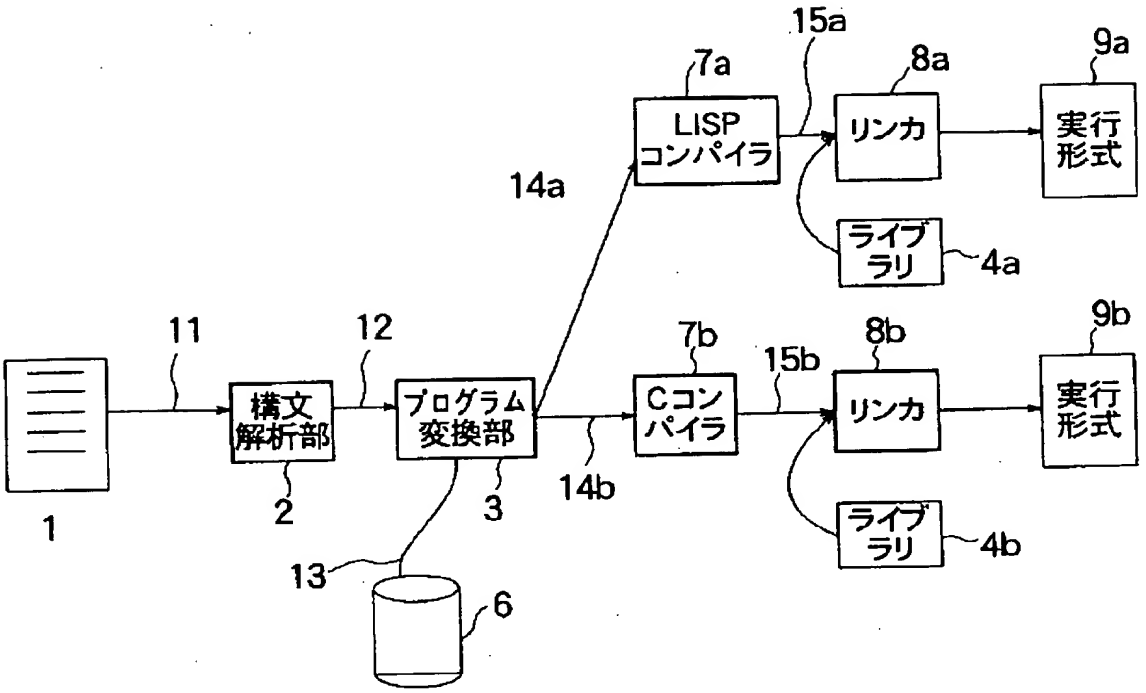
【図１】



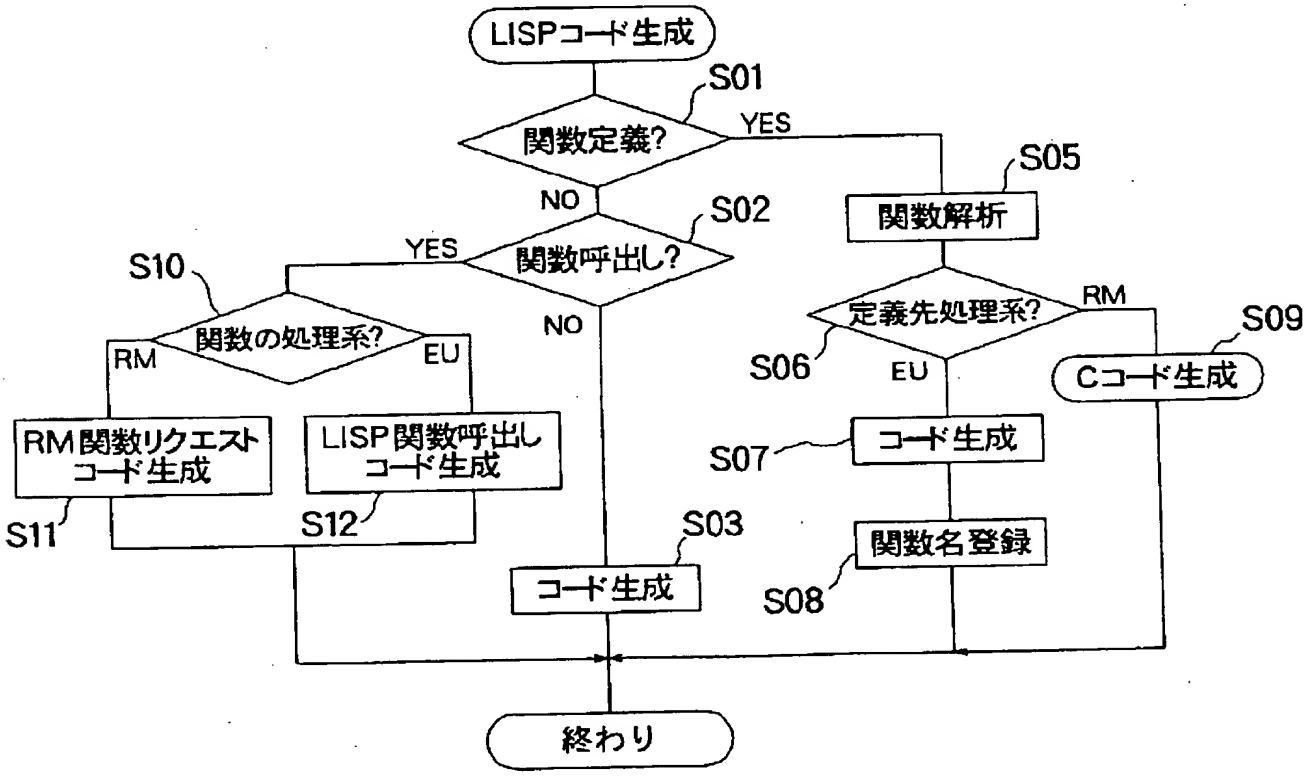
【図2】


ダイアグノスティクス
スタティクス
他のPE's
CRTディスプレイ
IBUS
EU
ALU
マスカ
シフタ
ローカル
メモリ
レジスタ
ファイル
ユーザ
スタック
システム
スタック
インストラクション
キャッシュ
システム
コントローラ
コントロール
メモリ
ディスパッチテーブル
ユーザスタック
RM I/F
MM I/F
ダイアグノスティクス
インターフェース
通信
インターフェース
RM
WS
EU I/F
ワーク
ステーション
ポインタ
マニュピレータ
二次記憶
リソース
ストレージ
MM

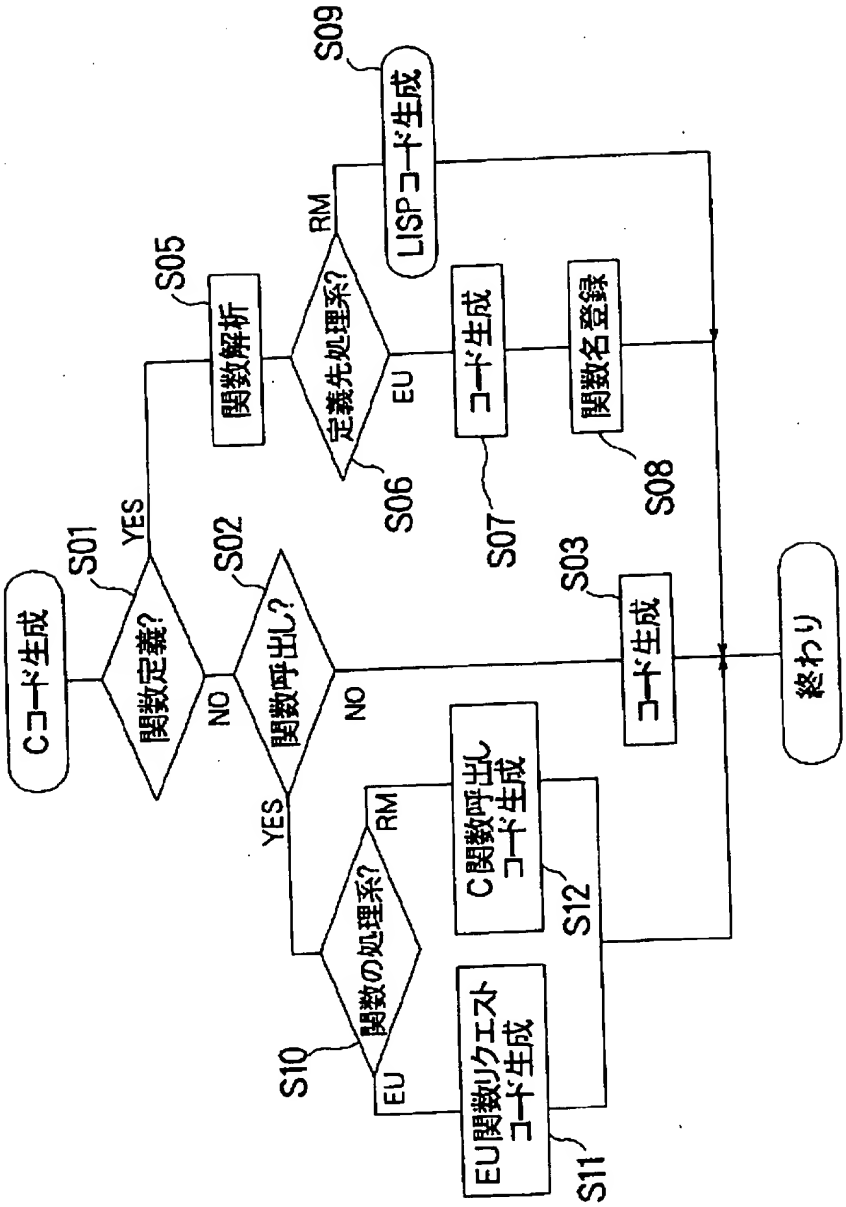
【図３】


15a
7a
8a
9a
LISP
コンパイラ
リンカ
実行
形式
14a
ライブ
ラリ
4a
11
12
7b
8b
9b
15b
構文
解析部
プログラム
変換部
Cコン
パイラ
リンカ
実行
形式
1
2
3
14b
ライブ
ラリ
4b
13
6

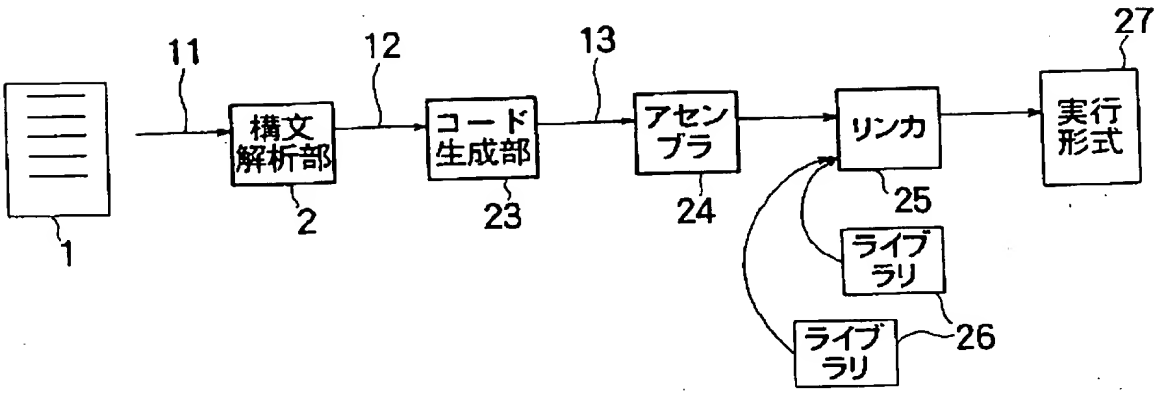
【図４】


LISPコード生成
S01
関数定義?
YES
NO
S02
関数呼出し?
YES
NO
S10
関数の処理系?
RM
EU
RM関数リクエスト
コード生成
LISP関数呼出し
コード生成
S11
S12
S03
コード生成
S05
関数解析
定義先処理系?
RM
EU
S06
S09
Cコード生成
S07
コード生成
関数名登録
S08
終わり

【図５】


Cコード生成
関数定義?
YES
NO
S01
関数呼出し?
YES
NO
S02
関数解析
S05
定義先処理系?
RM
EU
S06
LISPコード生成
S09
コード生成
S07
関数名登録
S08
コード生成
S03
終わり
関数の処理系?
RM
EU
S10
C関数呼出し
コード生成
S12
EU関数リクエスト
コード生成
S11

【図６】


11
12
13
27
構文解析部
コード生成部
アセンブラ
リンカ
実行形式
1
2
23
24
25
ライブラリ
ライブラリ
26